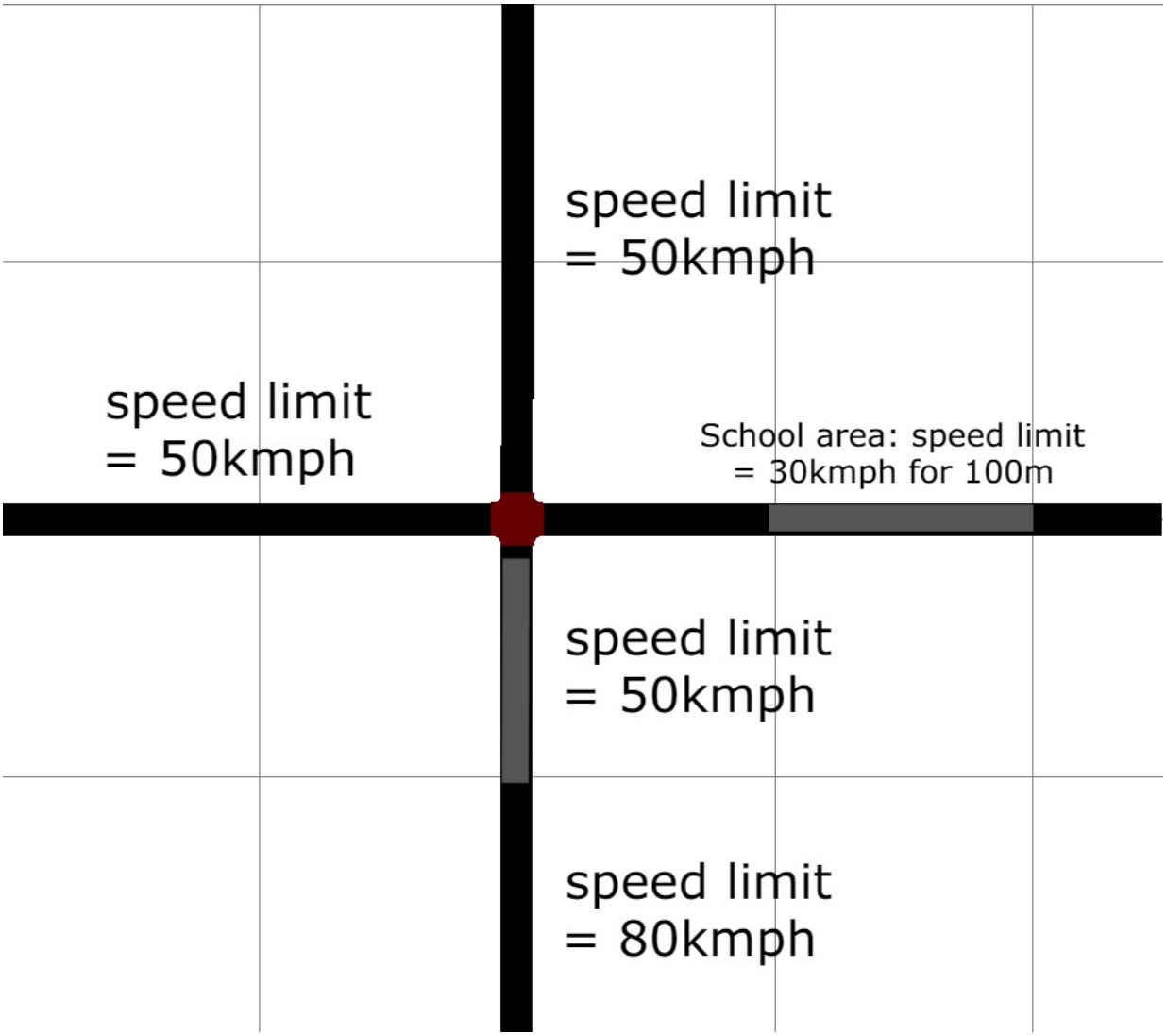
Weekly Updates (16/11/23) – Vinayak Gajendra Panchal

Driver type Classification - Unsupervised

Simulation setup:

Step-length	0.10 sec (10 cycles/sec)
Collision.action	remove
Collision.stoptime	20 sec
Collision.check-junctions	True
'- - log' (simulation log files)	simulation_main.log
Number of Vehicles	1000

2-Lane 4-Way Junction Details:



I have incorporated lanes with speed restrictions and a designated school zone area with limits ranging from 20-30 kmph (grey highlighted area). These speed restrictions have been implemented to enhance the replication of traffic scenarios within the LaneLet2 framework. A total of 1,000 vehicles were simulated for 10,000 seconds at a time-step length of 0.1 seconds. This resulted in a

comprehensive dataset of 700K FCD data points and 27 distinct attributes. The direct usage of this dataset might not be well-suited for clustering tasks, because there's a high likelihood that vehicles of different types might be grouped into the same clusters. This could happen due to the matching of their attributes at specific time stamps. For example, if two different types of vehicles (aggressive and violator) exhibit similar speed, acceleration, or other measurable attributes at the same time, the clustering algorithm might fail to distinguish between these different types, leading to inaccurate or misleading groupings.

To make this dataset suitable for clustering task, I aggregated the data to capture key features of each vehicle:

1. **Total Vehicle Time on Road**: I calculated the total active time for each vehicle by finding the difference between maximum and minimum timestamps.
2. **Spatial and Movement Aggregation**: Mean values of coordinates, angle, speed, position, and acceleration were computed, providing an overview of each vehicle's typical movement and location.
3. **Fuel and Distance**: Average fuel consumption and total distance covered were determined, indicating each vehicle's fuel efficiency and mobility range.
4. **Signals and Collisions**: The most common signal by vehicle and if the specific vehicle was involved in any type of collision were derived (1 if collision, 0 in no collision).
5. **Driver Type**: For each driver type was extracted from the original dataset. This will not used in the Clustering task but will be helpful for comparing the clusters.
6. Redundant columns were removed, resulting in a dataset better suited for clustering by highlighting distinguishing features and behaviors of the vehicles. This dataset has 1,000 data points corresponding to 1,000 vehicles in the simulation and 13 vehicle attributes.

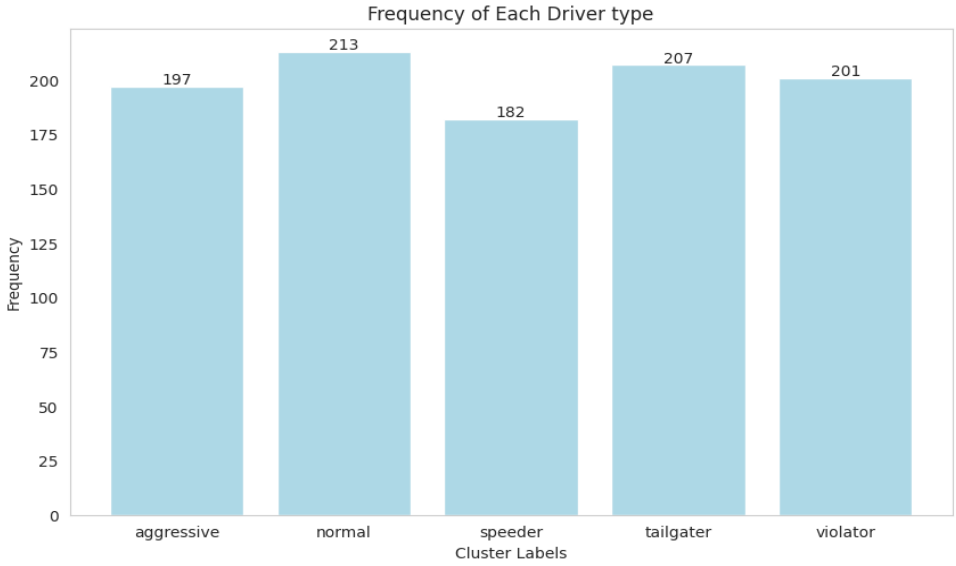


Figure 1. Frequency of True labels of Driver types

Following Unsupervised Learning models were considered for the study:

1. **KMeans Clustering**
2. **Hierarchical Clustering**
3. **Autoencoders – KMeans Clustering**

The following features from the dataset were considered for clustering of driver type:

total_time, avg_speed, avg_acceleration, distance_covered, max_signals, avg_fuel_consumed, collision_occurred.

Other features like *avg_x, avg_y, avg_pos,* and *avg_angle* were not selected as it wasn't relevant for clustering the driver type. To visualize the clusters in 2-dimension we need to reduce the dimensions of features, for that I used PCA and t-SNE to 2 components.

Results and Discussion:

1. KMeans Clustering:

The KMeans clustering algorithm was applied to group the data into 5 clusters which resulted in an inertia of 2080.15, it suggests that within-cluster variance is relatively high and silhouette score of 0.4163 which is moderately strong, indicating that on average, data points are closer to their own cluster, signifying fairly well-separated clusters.

Figure 2 Comparing frequencies of clusters with true labels.

The frequency histograms for each type of driver in the KMeans clusters show a big difference in the number of drivers in each cluster. Cluster 0 has the most drivers, which might mean that the behaviors in this cluster are more common, or maybe the clustering method prefers this cluster. Cluster 2 has only 74 drivers, which is a lot less than any driver type in the original labels. This might be because this driver type shares features with other types, causing its numbers to decrease. When we compare these clusters with the original labels, they don't match up very well. This suggests that KMeans might not have done a great job in correctly grouping the different types of drivers.

The PCA (Principal Component Analysis) scatter plots show a clear linear separation between clusters, capturing the most variance in the data. However, the spread of points within clusters, suggests that there might be sub-groups within these clusters. Compared to the Original Labels PCA plot we can see a few similarities in clusters.

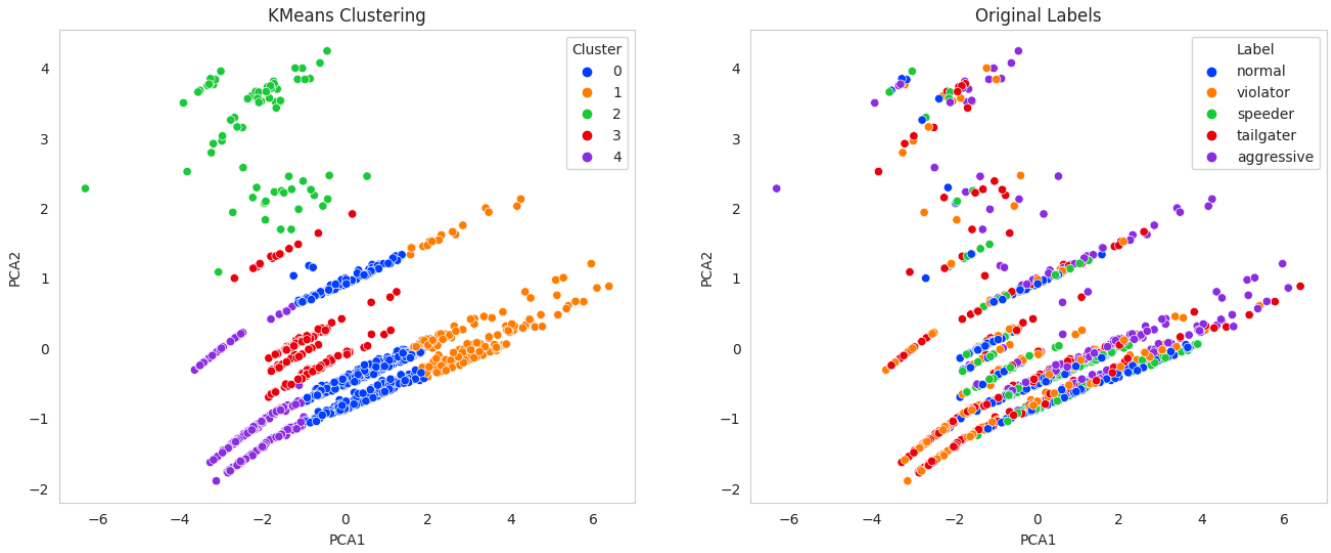


Figure 3 PCA Plot comparison of KMeans and true labels

The t-SNE (t-Distributed Stochastic Neighbor Embedding) visualization further supports the existence of well-defined clusters, particularly for clusters 1, 2, and 3, which seem to be more tightly grouped and better separated from each other compared to the PCA visualization.

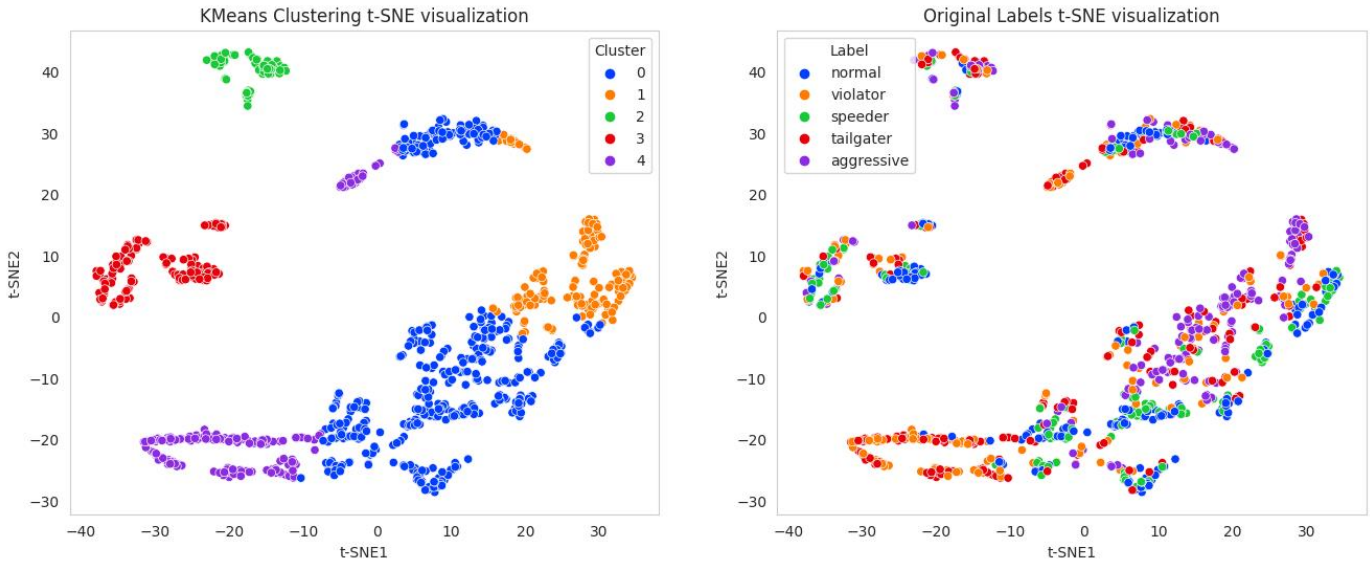


Figure 4 t-SNE Plot comparison of KMeans and true labels

2. Hierarchical Clustering:

The Hierarchical Clustering algorithm was applied to the dataset, resulting in 5 clusters when a distance value of 35 is used as the threshold for cutting the dendrogram. The dendrogram provides a visual representation of the clustering process, hierarchical structure in the data, and showing the levels at which individual clusters are merged. The silhouette score of 0.4081 is very close to the score obtained from KMeans clustering, which indicates that the clusters are also reasonably well separated in this model.

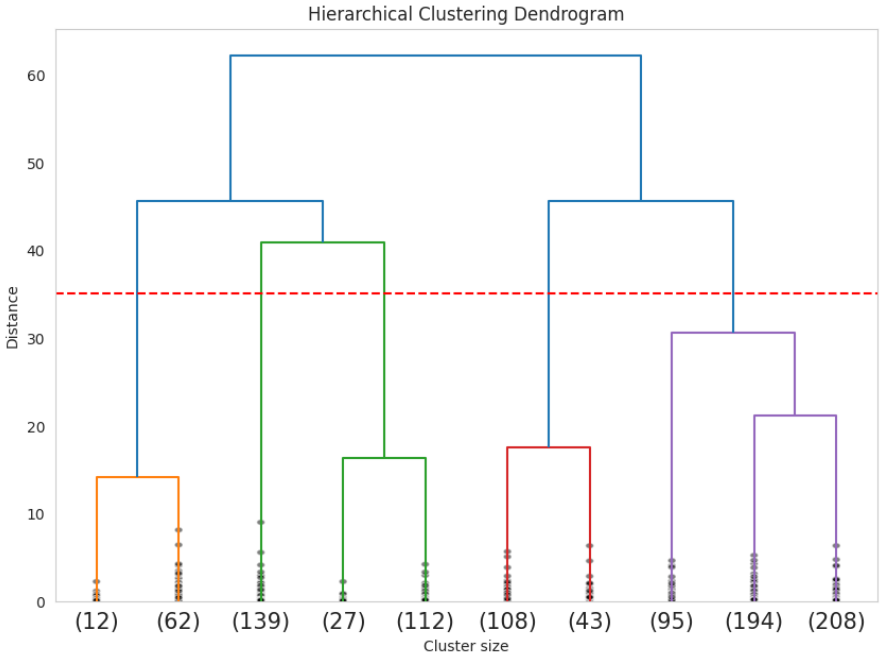


Figure 5 Dendrogram with horizontal cut at 35 for 5 clusters

In the frequency distribution histograms, we can observe a similar trend to the KMeans results, with one cluster (cluster 5) being much larger than the others. This could again suggest a prevalent driver behavior type or a potential aggregation of multiple behaviors within a single cluster. We can also see similar clusters frequency repetition from KMeans (cluster 1: 74, cluster 2 and 3: 139).

Figure 6 Comparing frequencies of clusters with true labels.

The PCA scatter plots and frequency bar chart for Hierarchical Clustering show that the clusters are less linearly separated compared to KMeans, with overlaps between some of the clusters.

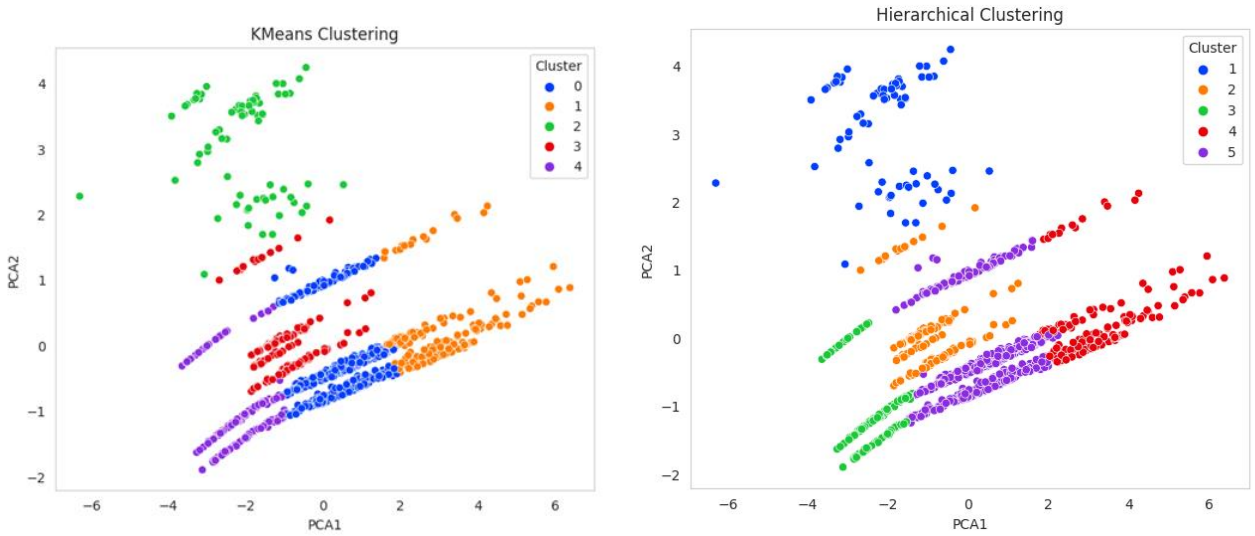


Figure 7 PCA Plot comparison of Hierarchical and KMeans

The t-SNE visualization for Hierarchical Clustering presents a more segregated arrangement of clusters. Some clusters appear to be well-defined, while others show a greater degree of mixing. This pattern might indicate that while some driver behaviors form distinct groups, others are not as easily separable and may share characteristics across groups.

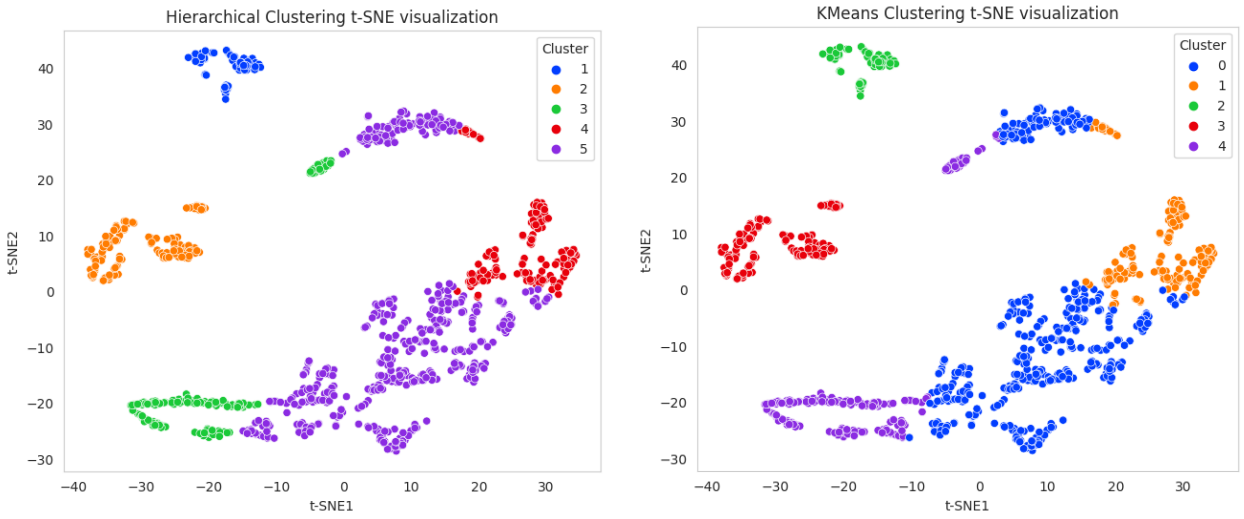


Figure 8 t-SNE Plot comparison of Hierarchical and KMeans

The silhouette score, which is slightly lower than that of the KMeans clustering, along with the frequency of each cluster, might imply that some cluster boundaries are overlapping in the Hierarchical model. However, the close similarity of silhouette scores between the two models suggests that the overall structure of the data is consistent, regardless of the clustering technique used.

3. Autoencoder - KMeans Clustering:

The Autoencoder-KMeans clustering approach has an encoder neural network architecture to reduce dimensionality (3 neurons) before applying KMeans clustering. The architecture consists of an input layer, followed by two layers of 5 neurons that compress the data to a lower dimension, and then expand it back to its original dimensionality. The model has a total of 180 parameters and a relatively small training loss of 0.1711, indicating good reconstruction performance by the autoencoder.

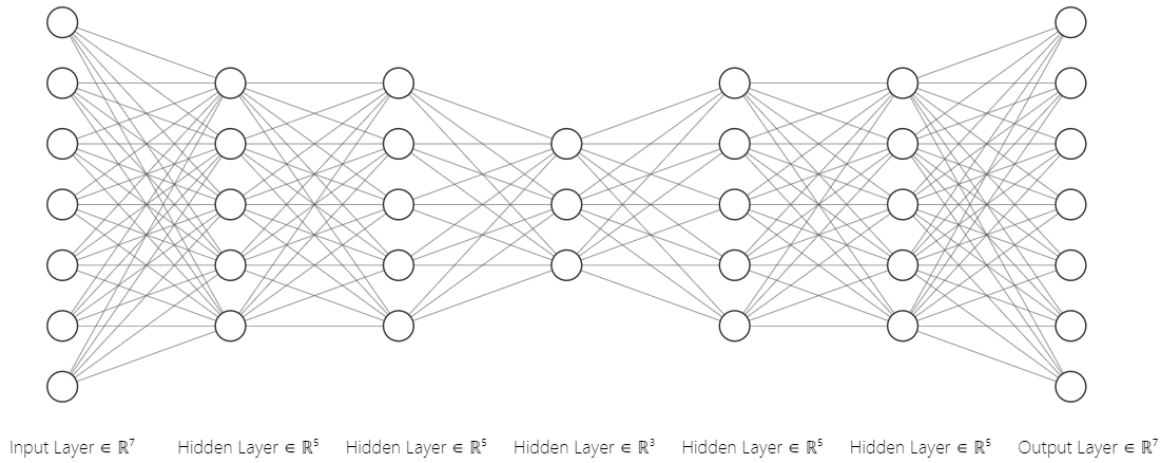


Figure 9 FCNN style of our Autoencoder for Clustering

The KMeans clustering applied after the encoder has resulted in an inertia of 981.2171, which is significantly lower than the inertia values obtained from KMeans clustering directly applied to the data. This lower inertia suggests that the autoencoder has created a more compact representation of the data, leading to tighter clusters. The silhouette score of 0.5891 is quite higher than those obtained from the KMeans and Hierarchical clustering methods, indicating that the clusters are denser and more well-separated.

The frequency distribution bar chart after autoencoder clustering shows that the distribution of data points across clusters is imbalanced, with cluster 2 being the largest. This could either reflect the actual distribution of driver behaviors in the dataset or indicate a bias in the clustering process towards certain patterns that are more in the reduced feature space.

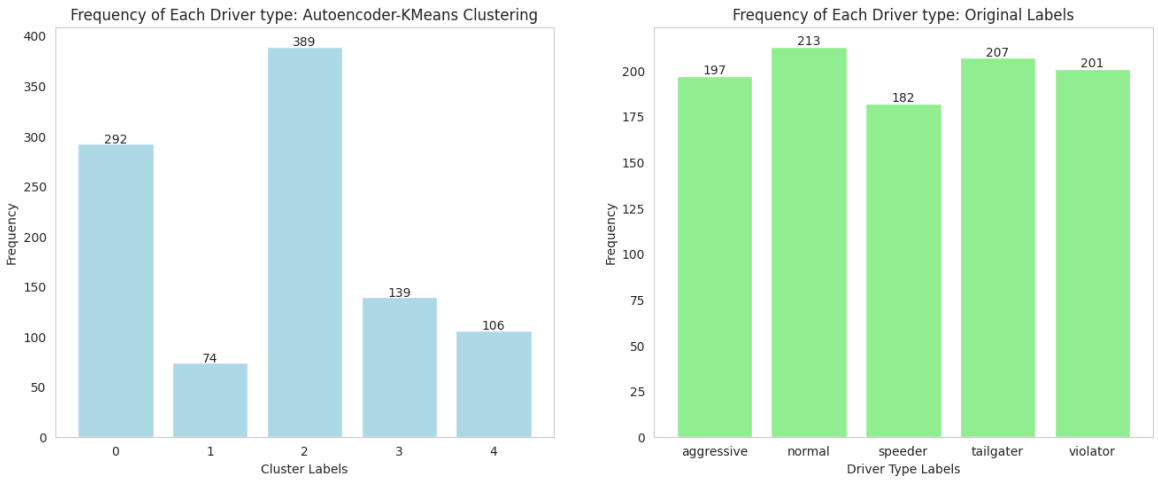


Figure 10 Comparing frequencies of clusters with true labels.

The PCA and t-SNE visualizations after autoencoder clustering show distinct clusters with minimal overlap. These visualizations support the quantitative measures, suggesting that the autoencoder-KMeans clustering method has effectively captured the underlying structure of the data.

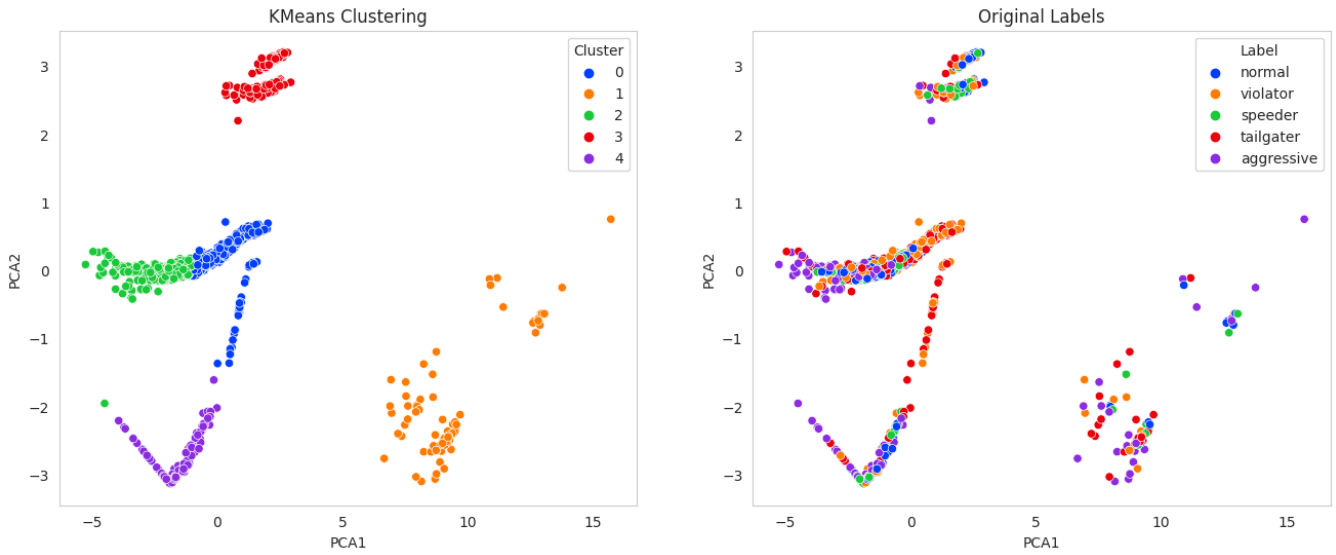


Figure 11 PCA Plot comparison of Autoencoder-KMeans and true labels

The autoencoder's encoder has effectively transformed the data into a space where KMeans clustering could be applied more effectively, resulting in well-defined and separable clusters that correspond to different driver types.

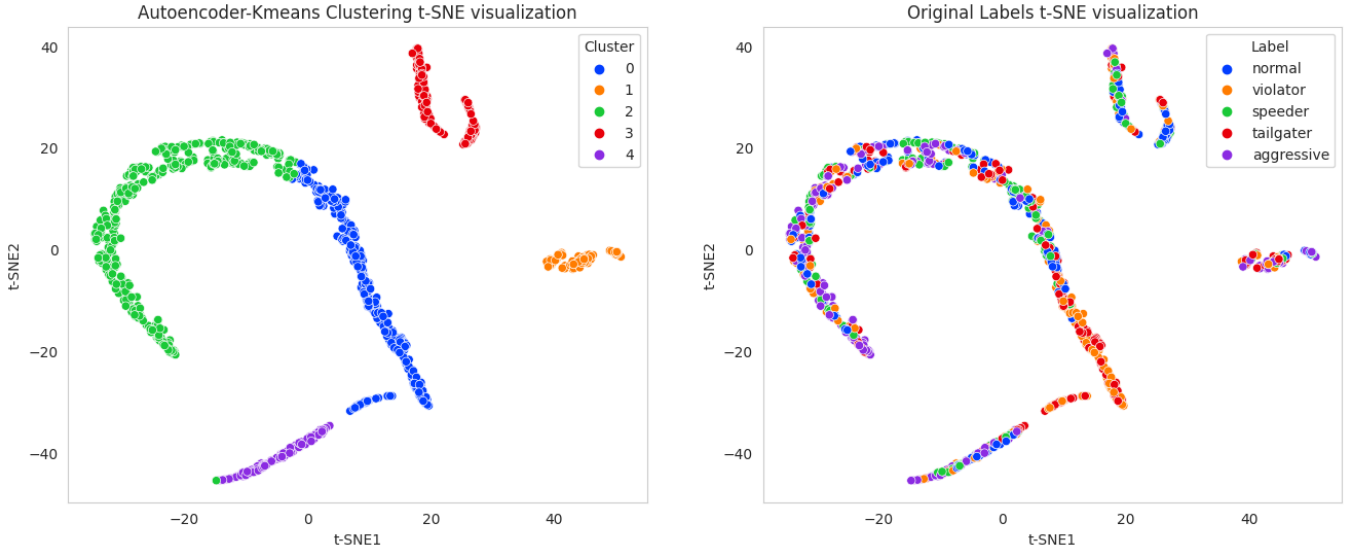


Figure 12 t-SNE Plot comparison of Autoencoder-KMeans and true labels

Overall Comparison:

The Autoencoder-KMeans clustering yielded the best results, with the lowest inertia indicating tighter and more compact clusters, and the highest silhouette score indicating better separation between clusters. KMeans clustering gave moderate separation but with a higher within-cluster variance

(inertia). Hierarchical clustering showed slightly less distinct clusters as per its silhouette score, frequency and PCA plot. From the bellow frequency plot, we can say that all clustering methods seem to identify one or two clusters that contain a disproportionately large number of data points. I believe that the large number is due to the matching of attributes with 'normal' vehicle type like speed and acceleration. The Autoencoder-KMeans has a more balanced distribution but still shows one cluster with a notably higher frequency compared to other clustering algorithms. We can also notice that all clustering algorithms have a common cluster frequency of 74 and 139 data points. All in all, there is an imbalance in clusters compared to the original label.

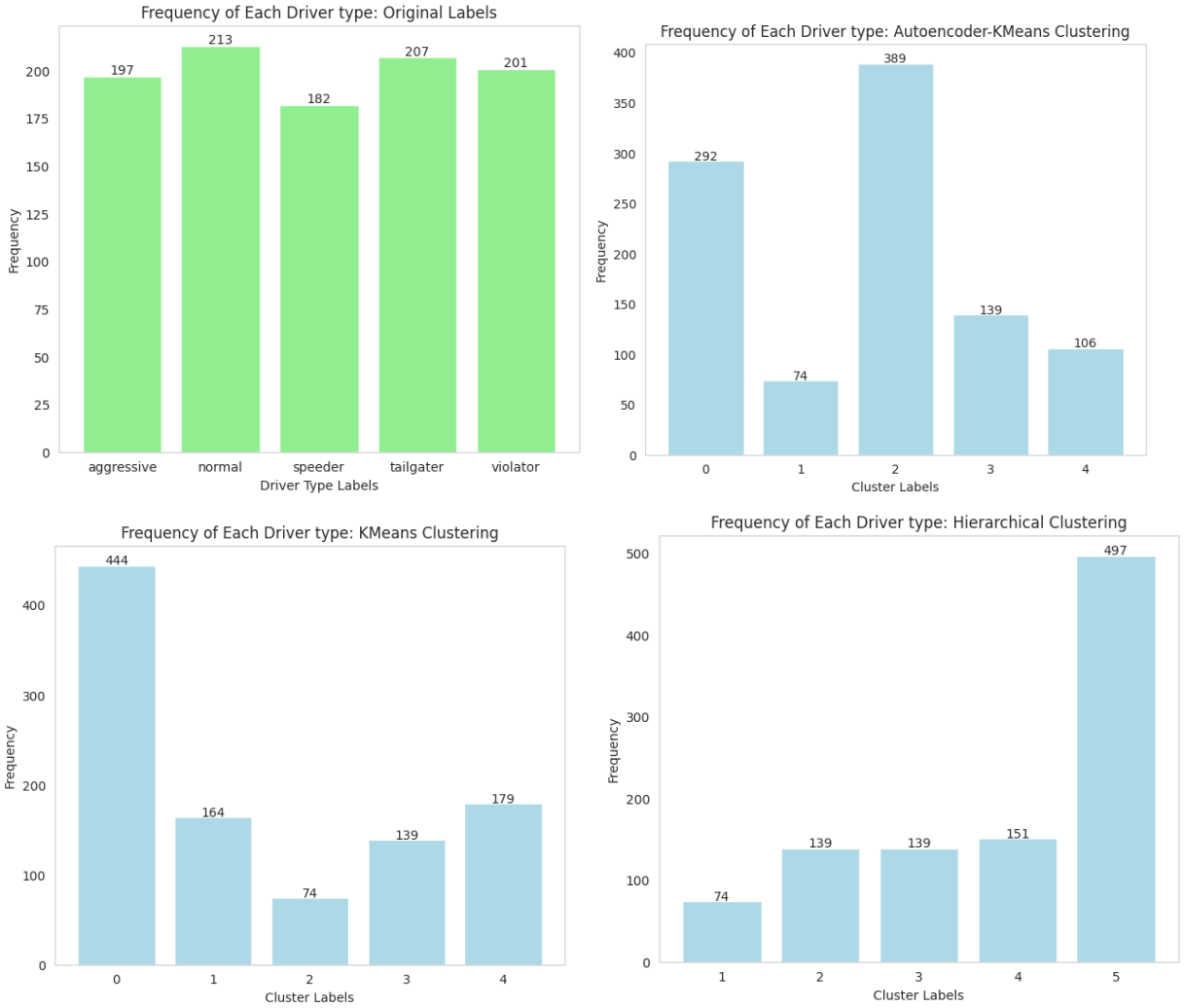


Figure 13 Frequency comparison of different clustering algorithms with original label